Fine penetration tests for fine websites

Dr.-Ing. Mario Heiderich, Cure53
Wilmersdorfer Str. 106
D 10629 Berlin
cure53.de · mario@cure53.de

Audit-Report Tuum Hedera Wallet Snap & Codebase 09.2024

Cure53, Dr.-Ing. M. Heiderich, Dr. D. Bleichenbacher, Dr. N. Kobeissi

Index

Fine penetration tests for fine websites

Dr.-Ing. Mario Heiderich, Cure53
Wilmersdorfer Str. 106
D 10629 Berlin
cure53.de · mario@cure53.de

Introduction

"Hedera Wallet Snap unlocks wallet functionality for Hedera via MetaMask that any other apps can interact with, thereby turning MetaMask into a native Hedera wallet without relying on Hedera JSON-RPC Relay. With Hedera Wallet Snap, users can send HBAR to another HBAR account id and an EVM address, retrieve account info from either the Hedera Ledger node or Hedera Mirror node."

From https://docs.tuum.tech/hedera-wallet-snap

This report (ID *TUU-05* for reference) presents the methodology and findings of a Cure53 penetration test and source code audit against the Tuum Hedera Wallet Snap and codebase.

In context, the Tuum Hedera Wallet Snap and codebase have been audited by Cure53 in two preceding audits, which were performed in November 2023 (see *TUU-02*) and in April 2024 (see *TUU-03*). This particular engagement was requested by Tuum Technologies, Inc. in July 2024 and carried out in CW39 September 2024.

The three-person testing team was tasked with determining the premise's current level of security protection, considering that new functionalities and alterations have been introduced between versions 0.3.1 and 0.6.0. To achieve this, the client invested a total of four work days within the budget and a single Work Package (WP) entitled *WP1: Pen.-tests & code audits against Tuum Hedera Wallet Snap & codebase* was created in advance.

The methodology adopted was white-box. To adhere with this approach, Cure53 was granted access to sources, test-supporting documentation, and a number of other useful assets. Preliminary actions to ensure a seamless segue into the active examination phase were completed by Cure53 in the build up to the procedures, specifically in CW38.

Cross-team communications were facilitated via Slack as usual, allowing the members of each organization to discuss the general progress, relay any interesting findings, and ask any clarifying questions if needed. The collaboration process was perfect and the reviews themselves were not delayed by any hindering factors, due to the exhaustive scope preparations. Live reporting was offered as a complementing tool but ultimately deemed surplus to requirements for this test iteration.

Despite ample evaluation depth, the Cure53 team was only able to identify two security relevant pitfalls, both of which were categorized as *Miscellaneous Issues* due to their *Low* associated risk.

Dr.-Ing. Mario Heiderich, Cure53
Wilmersdorfer Str. 106
D 10629 Berlin
cure53.de · mario@cure53.de

Evidently, the negligible yield of issues reflects highly favorably on the effectiveness of the developer team's implementation compared to the prior audit. To corroborate this positive viewpoint, one of the tickets was deemed out-of-scope and the other should be relatively straightforward to address.

Onward, the report is divided into a number of key chapters for ease of reference. Firstly, the *Scope* provides all general setup information in concise bullet points. Next, the *Test Methodology* clarifies the evaluation techniques applied by Cure53 and all interesting subsequent observations. This section hopefully verifies the extent of the test team's endeavors, despite the low yield of findings. The document then lists all security problems in chronological order of detection (rather than degree of severity), grouped into two subcategories: *Identified Vulnerabilities* (albeit none were detected here) and *Miscellaneous Issues*. Each corresponding ticket gives a technical explanation, Proof-of-Concept (PoC) and/or steps to reproduce, code snippets, and remedial advice. Lastly, the *Conclusions* section summarizes Cure53's overarching viewpoints of the core focus elements, as well as substantiates their security effectiveness.

Dr.-Ing. Mario Heiderich, Cure53
Wilmersdorfer Str. 106
D 10629 Berlin
cure53.de · mario@cure53.de

Scope

- **Code audits & security reviews against Tuum Hedera Wallet Snap & related codebase**
  - **WP1:** Pen.-tests & code audits against Tuum Hedera Wallet Snap & codebase
    - **Sources:**
      - https://github.com/hashgraph/hedera-metamask-snaps/compare/2d164945740a2615ad9eeb6e5ea6c234ac18c873...47780fdaa69e637e8ddce2d5bb1b2fc7a8c3cfae
    - **Commit:**
      - Comparison between:
        - v0.3.1: d164945740a2615ad9eeb6e5ea6c234ac18c873
        - v0.6.0: 47780fdaa69e637e8ddce2d5bb1b2fc7a8c3cfae
    - **Primary focus:**
      - Evaluate all differences between v0.3.1 and v.0.6.0.
      - Ensure regressions in functionality or security have not been introduced.
    - **Documentation:**
      - https://docs.tuum.tech/hedera-wallet-snap
  - **Test-supporting material was shared with Cure53**
  - **All relevant sources were shared with Cure53**

Dr.-Ing. Mario Heiderich, Cure53
Wilmersdorfer Str. 106
D 10629 Berlin
cure53.de · mario@cure53.de

Test Methodology

The primary objective of this security audit was to evaluate the modifications introduced between versions 0.3.1 and 0.6.0 of the Hedera Wallet Snap for MetaMask. Cure53 sought to verify that the new functionalities and alterations have not exposed any security vulnerabilities, and that the Snap continues to operate securely and reliably within the MetaMask environment.

The audit encompassed a comprehensive review of all code changes, additions, and deletions between the two versions. This included:

- New features and functionalities added to the Snap.
- Modifications to existing code, including logic and structure changes.
- Updates to user interface components and corresponding interactions.
- New utility functions and validators.
- Alterations affecting cryptographic operations and key management.
- Reviews of documentation and comments for accuracy and security implications.

Methodology Overview

Cure53's audit methodology consisted of a multi-faceted approach combining automated tools and manual analysis to ensure thorough coverage. The specific steps included:

- **Differential Analysis**
  - Performed a file-by-file comparison between versions 0.3.1 and 0.6.0 to identify all changes.
  - Categorized changes into additions, deletions, and modifications for targeted review.
- **Static Code Analysis**
  - Utilized static analysis tools (e.g., ESLint with security plugins) to scan the codebase for common vulnerabilities such as injection flaws, insecure cryptographic practices, and data leakage.
  - Reviewed dependency changes using *npm audit* to identify any known vulnerabilities in third-party packages.
- **Manual Code Review**
  - Conducted a detailed manual review of all identified changes, focusing on security-critical areas such as input validation, cryptographic operations, and sensitive data handling.
  - Assessed the implementation of new features against security best practices and Hedera guidelines.
- **Functional Testing**
  - Executed the Snap within a controlled MetaMask environment to test new and existing functionalities.

Dr.-Ing. Mario Heiderich, Cure53
Wilmersdorfer Str. 106
D 10629 Berlin
cure53.de · mario@cure53.de

  - Performed end-to-end testing of key features, including HBAR transfers, account information retrieval, and interactions with Hedera services (Token Service, Smart Contract Service, and Consensus Service).
- **Security Testing**
  - Investigated input validation mechanisms, particularly in new functions such as the smart contract parameters validator in *HederaUtils.ts*, to ensure robust defense against malformed or malicious inputs.
  - Evaluated the handling and storage of private keys and other sensitive information, ensuring that they are securely managed within the Snap's execution environment.
  - Verified that cryptographic operations employ secure algorithms and optimal key management practices.
- **User Interface Review**
  - Inspected new and modified UI components in the *src/components* directory.
  - Scrutinized components that display sensitive information, such as those revealing private keys, ensuring that they employ secure rendering methods like *CopyableSensitive*.
  - Ensured that user prompts and notifications are clear and do not inadvertently encourage unsafe behavior.
- **Regression Testing**
  - Conducted tests to confirm that existing functionalities from version 0.3.1 remain unaffected and operate as expected in version 0.6.0.
  - Checked for any deprecated functions or backward compatibility issues that could affect users upgrading to the new version.
- **Compliance Verification**
  - Verified adherence to MetaMask Snaps security requirements and best practices.
  - Determined compliance with Hedera network protocols and security guidelines.

Review Process - Code Comparison & Analysis

- **Minor and Typographic Changes**
  - The files and commands itemized below were evaluated for minor and typographic changes:
    - *src/client/HederaClientImplFactory.ts*
    - *src/client/SimpleHederaClientImpl.ts*
    - *src/commands/account/**
    - *src/commands/allowance/**
    - *src/commands/hts/**
    - *src/commands/StakeHbarCommand.ts*
    - *src/commands/TransferCryptoCommand.ts*
  - Cure53 verified that these changes did not affect the logic or introduce any vulnerabilities.

Dr.-Ing. Mario Heiderich, Cure53
Wilmersdorfer Str. 106
D 10629 Berlin
cure53.de · mario@cure53.de

- **Hedera Consensus Service Commands (*src/commands/hcs/**)**
  - Reviewed constructors and transaction executors in the following areas:
    - *CreateTopicCommand.ts*
    - *DeleteTopicCommand.ts*
    - *SubmitMessageCommand.ts*
    - *UpdateTopicCommand.ts*
  - Ensured that topic creation, deletion, message submission, and updates are securely implemented with performant permission checks and input validation.
- **Hedera Smart Contract Service Commands (*src/commands/hscs/*)***
  - Assessed the integrated functionalities for secure contract interactions.
  - Checked for correct handling of smart contract calls, parameter passing, and response processing.
- **Sign Message Command**
  - Noted the removal of *src/commands/SignMessageCommand.ts*.
  - Confirmed that said removal had not impacted the Snap's security or functionality.
- **Snap Accounts (*src/snap/SnapAccounts.ts*)**
  - Evaluated new UI notifications and their potential impact on user experience and security.
  - Ensured that notifications do not expose sensitive information or mislead users.
- **UI Components (*src/components/**)**
  - Reviewed components for secure rendering practices, especially those dealing with sensitive data.
  - Verified that components utilize *CopyableSensitive* or equivalent methods to prevent unintended data exposure.

Review Process - Utility Functions & Validators

- **Cryptographic Utilities (*src/utils/CryptoUtils.ts*)**
  - Analyzed new utility functions for cryptographic operations.
  - Confirmed the use of secure algorithms and effective error handling.
- **Smart Contract Parameters Validator (*src/utils/HederaUtils.ts*)**
  - Performed an in-depth review of the newly introduced validator.
  - Tested with various input scenarios to ensure sound validation and prevention of injection attacks.
- **Transaction Record Formatting (*src/utils/Utils.ts*)**
  - Checked the formatting utilities for potential data leakage or improper handling of transaction data.

Dr.-Ing. Mario Heiderich, Cure53
Wilmersdorfer Str. 106
D 10629 Berlin
cure53.de · mario@cure53.de

Miscellaneous Issues

This section covers any and all noteworthy findings that did not incur an exploit but may assist an attacker in successfully achieving malicious objectives in the future. Most of these results are vulnerable code snippets that did not provide an easy method by which to be called. Conclusively, while a vulnerability is present, an exploit may not always be possible.

TUU-05-001 WP1: Unpatched packages utilized within wallet codebase *(Low)*

Cure53 observed that some libraries with known security vulnerabilities are utilized within the Hedera Wallet Snap complex. Whether these vulnerabilities are exploitable, however, depends on how and to what extent the relevant functionality is currently leveraged in the targeted application.

One must note that the testing team was unable to fully ascertain the impact of these packages, owing to time constraints and the need to focus on other priority areas. As a result, the wider implications are unclear and warrant further research from the in-house team as soon as possible.

Library name	Version
path-to-regexp	*2.0.0 - 3.2.0*
elliptic	*2.0.0 - 6.5.6*
@grpc/grpc-js	*<1.8.22*

In general, supply chain security can be challenging to provide to a first-rate standard and a definitive solution oftentimes cannot be offered. However, developers can conform with best practices in this area by ensuring that the most recent versions of each library are installed. This proactive strategy will help the construct to benefit from any patches and enhancements applied as a result of flaws or weaknesses identified in the past.

To mitigate this issue, Cure53 recommends upgrading all relevant libraries and enforcing the retention of up-to-date libraries via a periodic or automated internal strategy.

Dr.-Ing. Mario Heiderich, Cure53
Wilmersdorfer Str. 106
D 10629 Berlin
cure53.de · mario@cure53.de

TUU-05-002 OOS: Incorrect signature verification by the library elliptic *(Low)*

For this exercise, Cure53 briefly analyzed the underlying *elliptic* library used in the in-scope application by running it against a suite of test vectors. This procedure uncovered certain security problems.

Firstly, the verification of an ECDSA signature is expected to return a boolean value; specifically, *true* if the signature verifies and *false* upon failure. However, the audit team confirmed the ability to craft malformed ECDSA signatures that trigger an exception. If these exceptions are not caught by the application, the likelihood of Denial of Service (DoS) attacks increases.

Secondly, Cure53 determined that some valid ECDSA signatures are not accepted by the library. A meticulous evaluation of this fault is pending, though one can assume that these errors are caused by an arithmetic error. Pertinently, an exploit for this limitation is not available at present.

To mitigate this issue, Cure53 suggests contacting the author of the library following the confirmation that disclosing said weakness is appropriate.

Dr.-Ing. Mario Heiderich, Cure53
Wilmersdorfer Str. 106
D 10629 Berlin
cure53.de · mario@cure53.de

Conclusions

For this Q3 2024 assignment, Cure53 conducted an extensive security audit of the Hedera Wallet Snap, focusing on all changes introduced between versions 0.3.1 and 0.6.0, including new features, modifications, and deletions. The methodology entailed a varied approach to ensure diligent coverage over the targeted codebase, combining differential analysis, static code analysis, manual code reviews, functional and security testing, user interface evaluation, regression testing, and compliance verification. Key audit areas included an exploration of new functionalities in the Hedera Consensus Service and Smart Contract Service commands; an assessment of new utility functions and validators for secure input handling; UI components, especially those handling sensitive information like private keys; plus cryptographic operations and key management practices for security compliance.

In summary, Cure53 strove to verify whether any of the new additions incur unwanted side effects. For example, a common drawback with new features is that additions to the code can introduce type confusion attacks or fallback attacks, though fortunately no vulnerability of this ilk was detected. Modules that called cryptographic primitives or exhibited a notable number of alterations were of higher priority. These were reviewed independently of previous audits by pentesters with no prior experience handling the aspects in question. Despite these conditions, no associated flaws were located.

Elsewhere, Cure53 compared the implementation with typical industry practices. One deviation from BIP44 was detected whereby a derivation path failed to utilize hardened nodes, contrary to the BIP44 proposal. Albeit, the team confirmed that this deviation was intentional to provide backwards compatibility with existing legacy solutions. As such, the key management in this context is adequate, asserting that key material is not reused and potential faults with the legacy code would only affect a single coin type. Following the completion of the vetting procedures, Cure53 is pleased to verify that no new security vulnerabilities are present in the updated Hedera Wallet Snap version 0.6.0. The augmentations have been implemented in adherence to security best practices and comply with general MetaMask Snaps and Hedera guidelines.

TUU-05-001 pertains to a miscellaneous finding regarding unpatched NPM package dependencies. These packages and third-party libraries should be updated in order to improve the Snap's overall security posture and reduce vulnerability exposure. To provide closing advice, Cure53 recommends establishing protocols to ensure continuous monitoring and timely updating of dependencies, as well as to continue providing clear instructions to users on optimal security measures, particularly with respect to the handling of private keys and sensitive data.

Cure53 would like to thank Kiran Pachhai from the Tuum Technologies, Inc. team for his excellent project coordination, support, and assistance, both before and during this assignment.